the globus alliance
www.globus.org

UDT as an Alternative Transport Protocol for GridFTP

Raj Kettimuthu

kettimut@mcs.anl.gov

Argonne National Laboratory

The University of Chicago

the globus alliance
www.globus.org

Outline

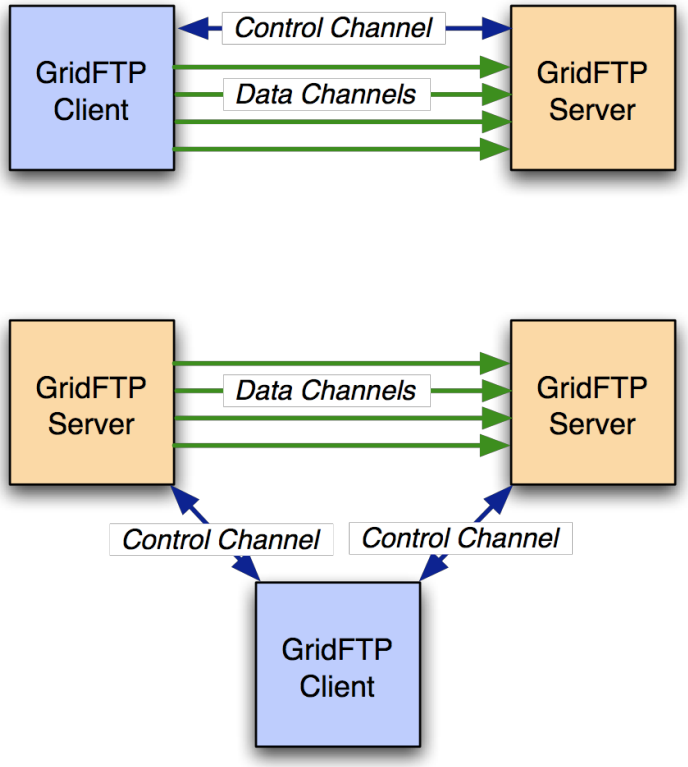
- GridFTP
- GridFTP Architecture
- Globus XIO
- UDT
- GridFTP/UDT integration
- Experimental results

GridFTP

- A secure, robust, fast, efficient, standards based, widely accepted data transfer protocol
- We also supply a reference implementation:
  - Server
  - Client tools (globus-url-copy)
  - Development Libraries
- Multiple independent implementations can interoperate
  - University of Virginia and Fermi Lab have home grown servers that work with ours.
- Lots of people have developed clients independent of the Globus Project.

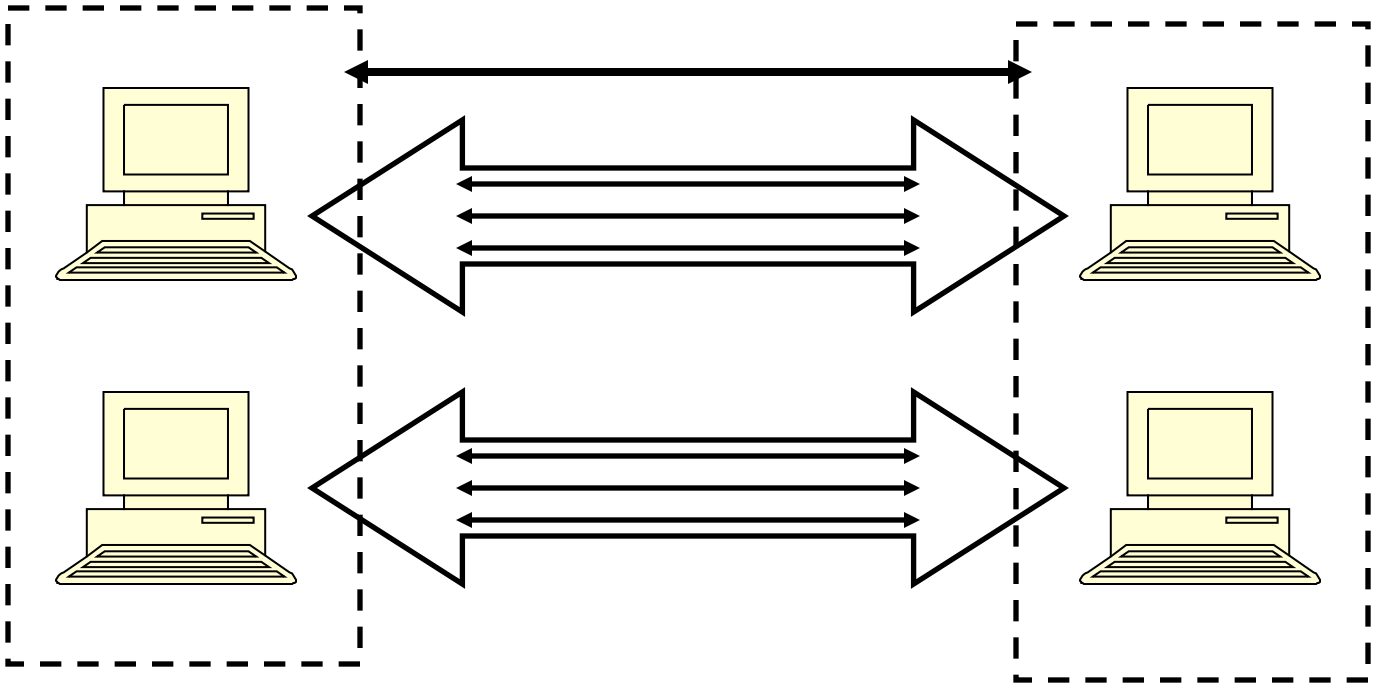
GridFTP

- Two channel protocol like FTP
- Control Channel
  - Communication link (TCP) over which commands and responses flow
  - Low bandwidth; encrypted and integrity protected by default
- Data Channel
  - Communication link(s) over which the actual data of interest flows
  - High Bandwidth; authenticated by default; encryption and integrity protection optional



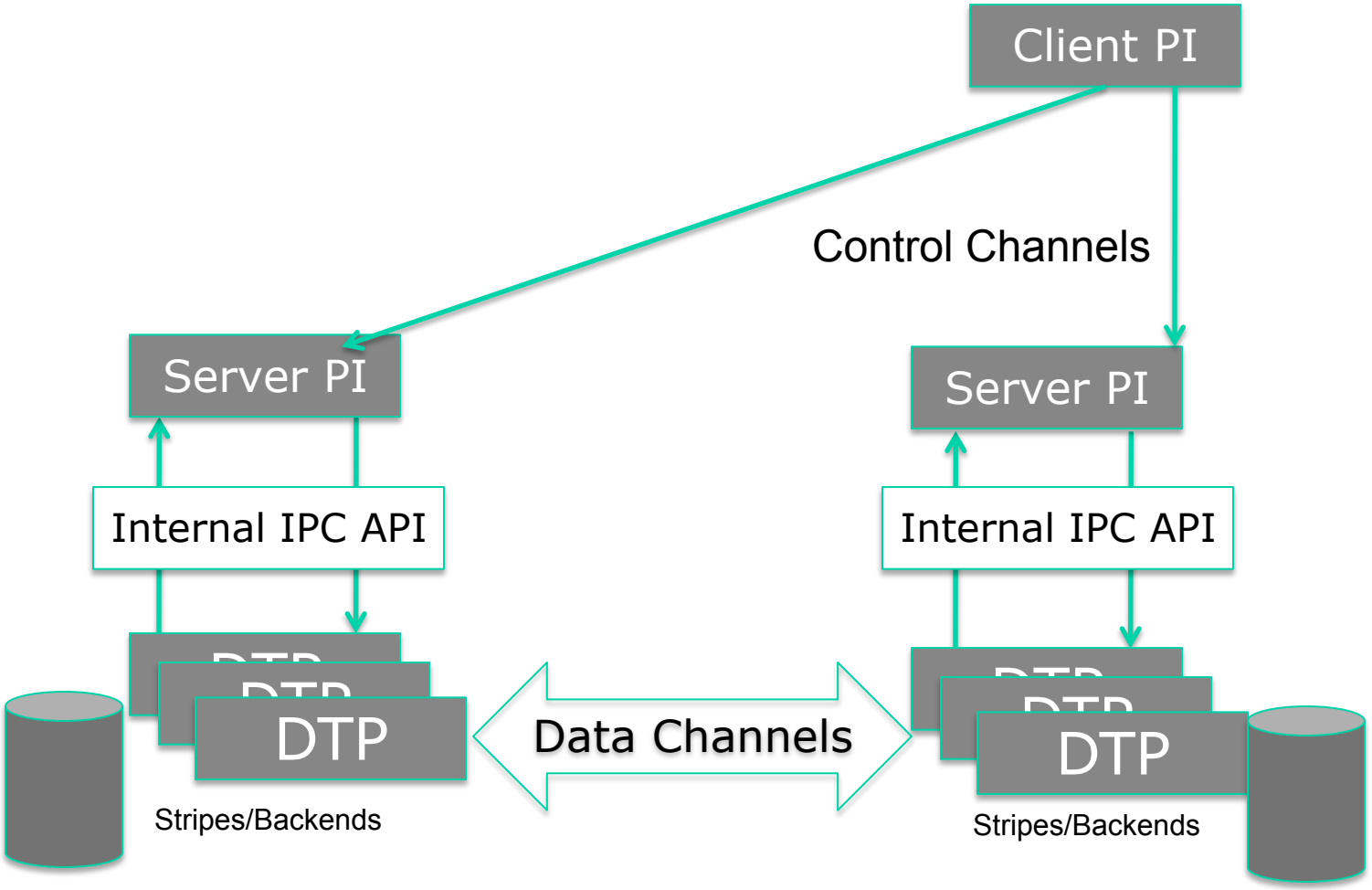
Striping

- GridFTP offers a powerful feature called striped transfers (cluster-to-cluster transfers)



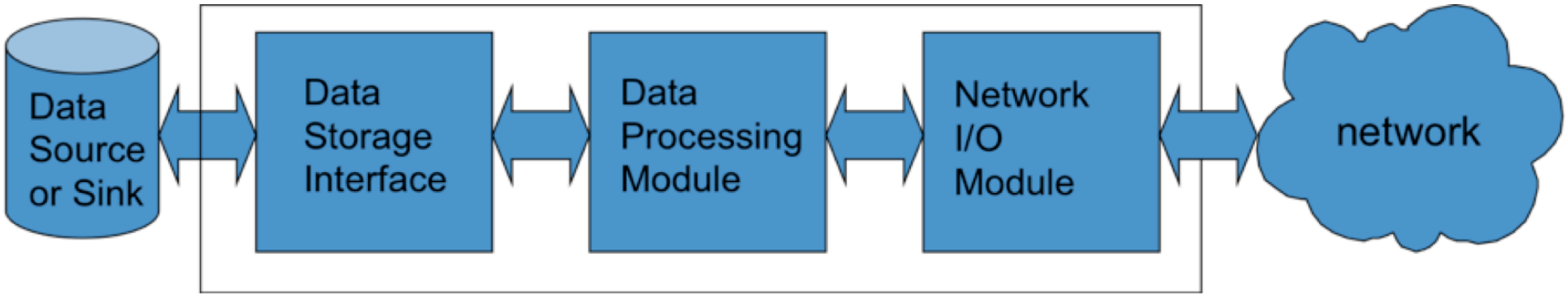
the globus alliance
www.globus.org

GridFTP Architecture



GridFTP Architecture

GridFTP Data Transfer Pipeline

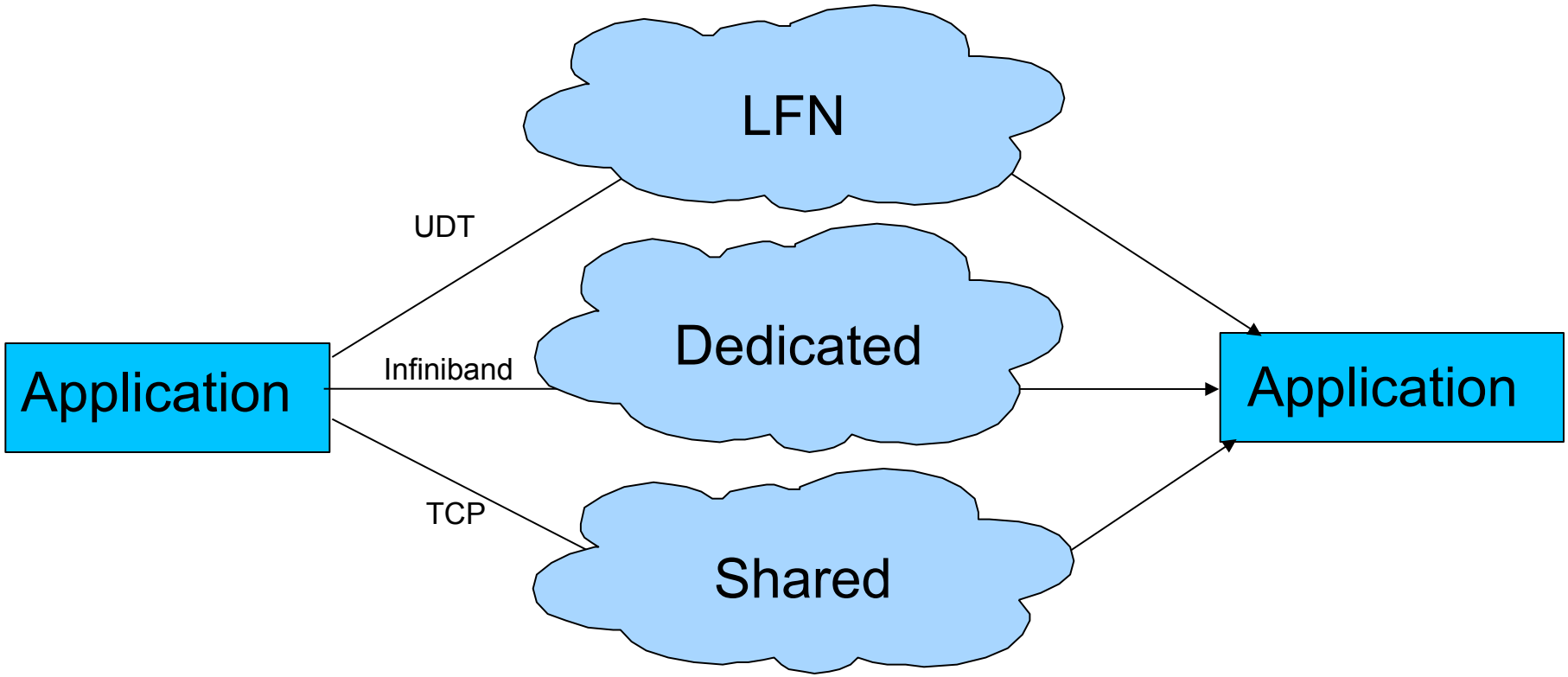


Globus XIO

Grid Communication

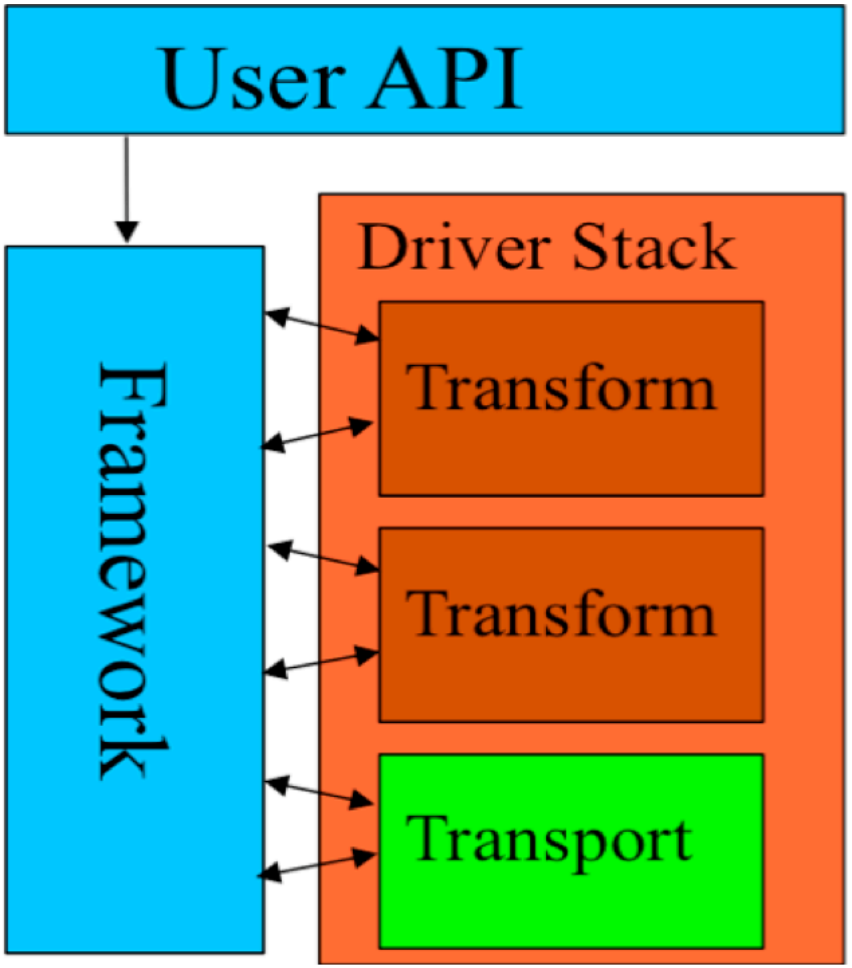
- Geographically Distributed Resources
- Varying Networks Characteristics
  - LAN, WAN, LFN, Dedicated, Shared, QOS
- Varying Network Protocols
  - HTTP, UDT, TCP, RBUDP, etc.
  - Researching making newer and faster
- Varying Conditions
  - Congested/Idle

Varying Environments

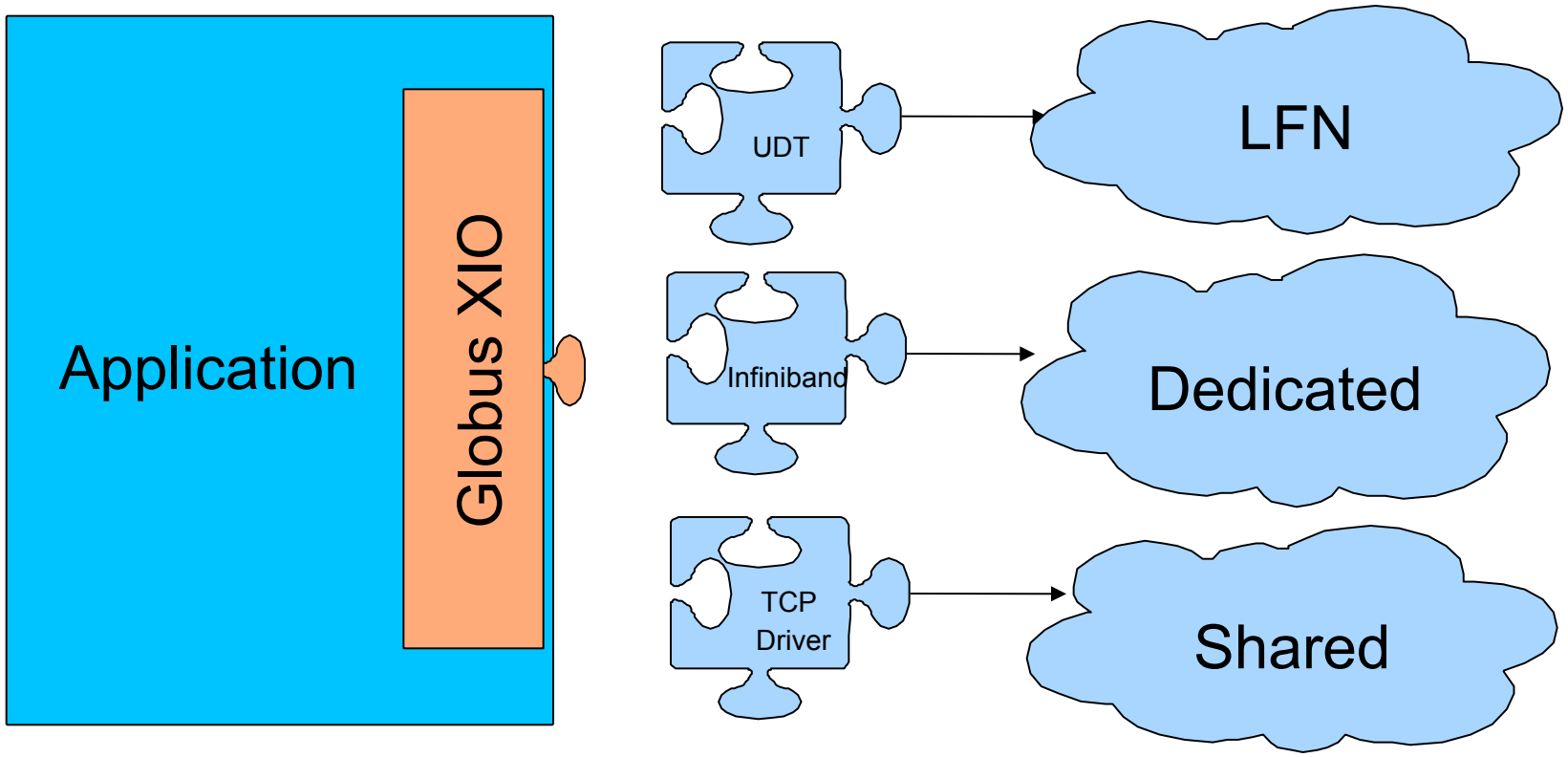


Globus XIO

- Framework to compose different protocols
- Provides a unified interface open/close/ read/write
- Driver interface to hook 3rd party protocol libraries

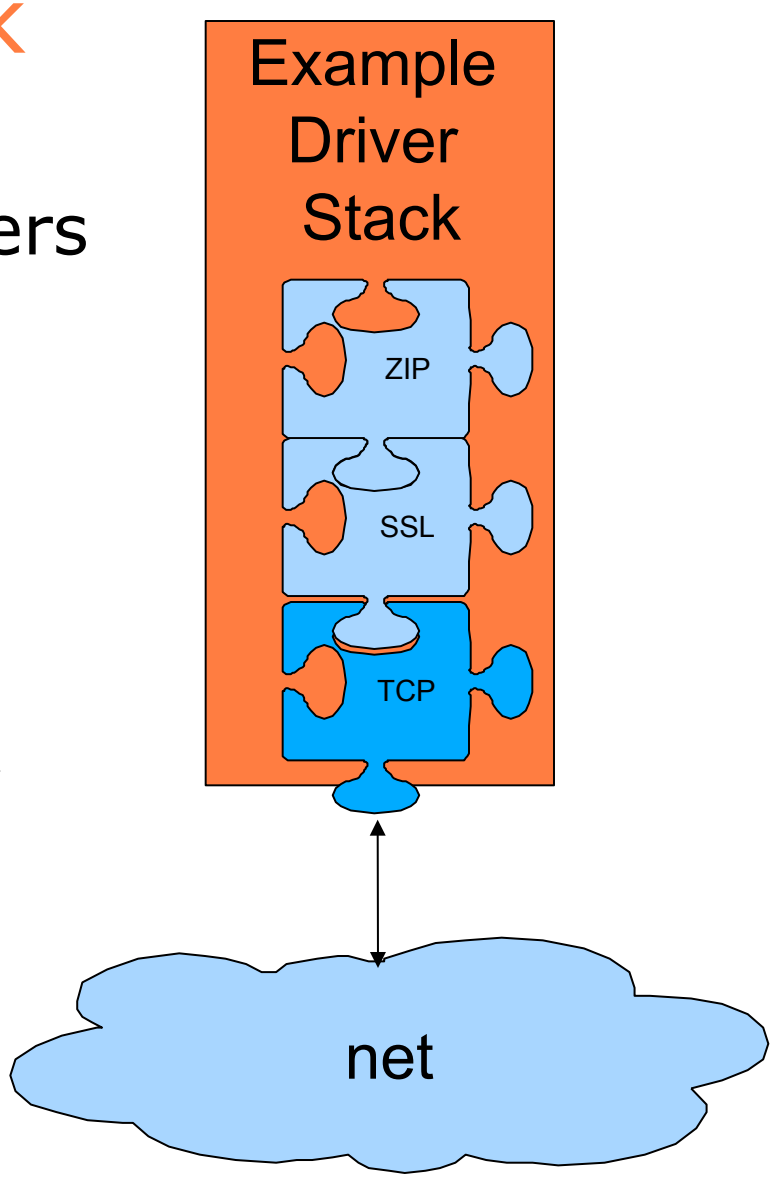


Varying Networks



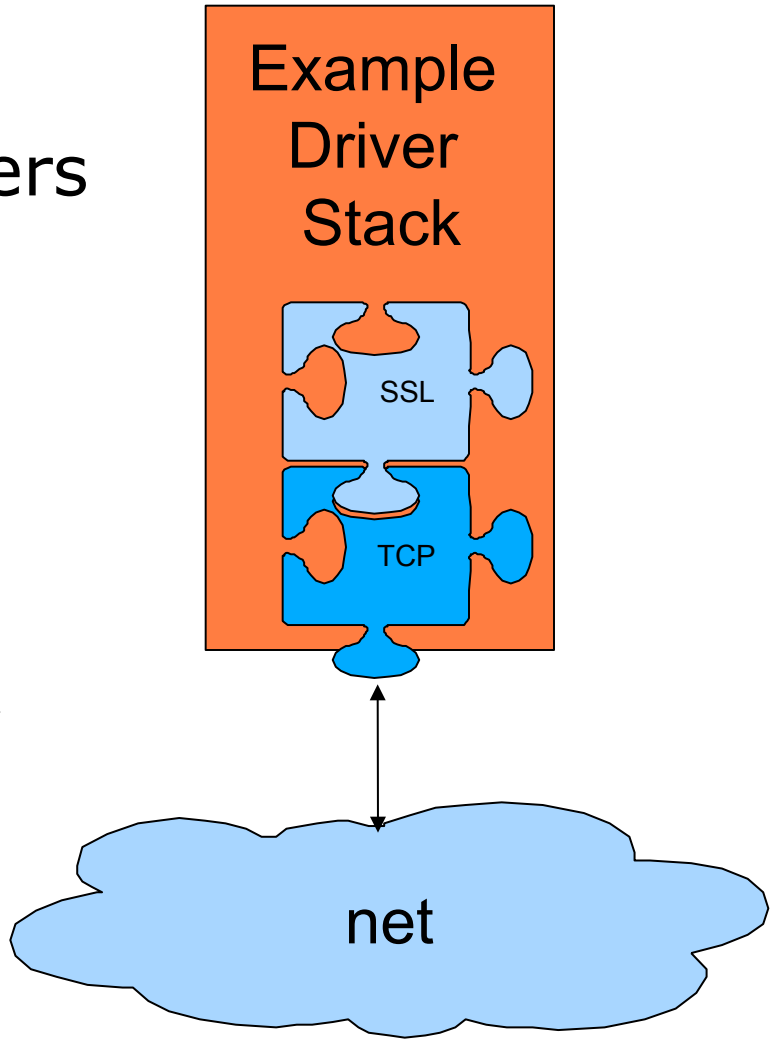
Stack

- An arrangement of drivers
- Transport
  - Exactly one per stack
  - Must be on the bottom
- Transform
  - Zero or many per stack



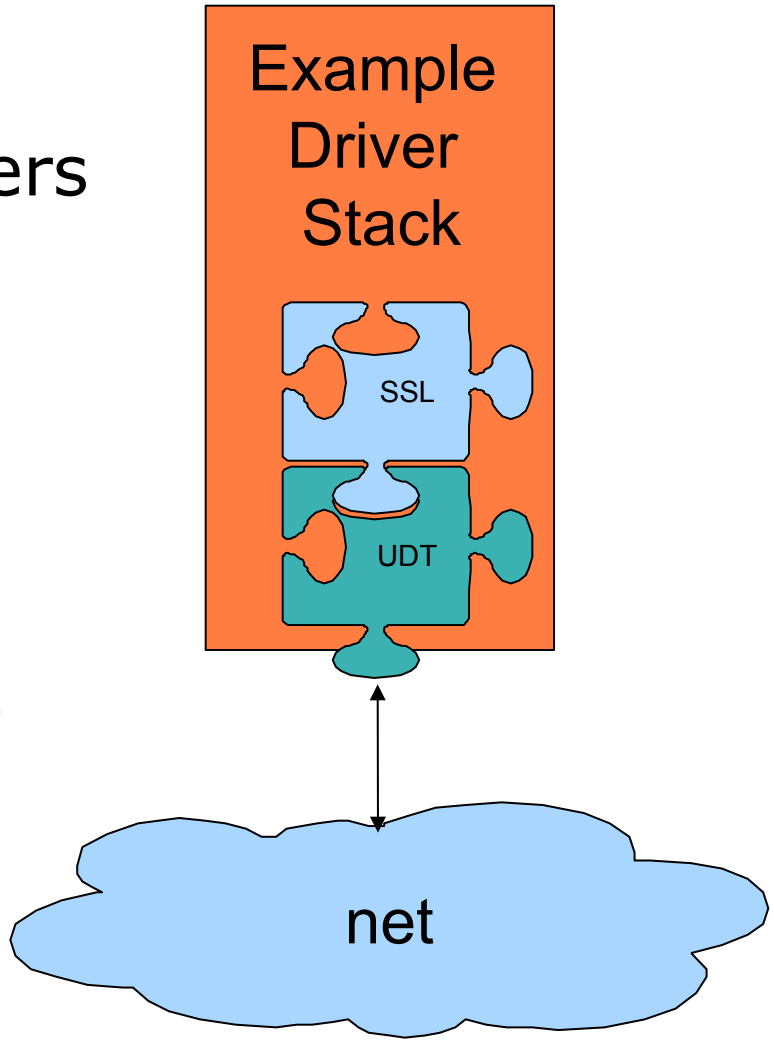
Stack

- An arrangement of drivers
- Transport
  - Exactly one per stack
  - Must be on the bottom
- Transform
  - Zero or many per stack



Stack

- An arrangement of drivers
- Transport
  - Exactly one per stack
  - Must be on the bottom
- Transform
  - Zero or many per stack



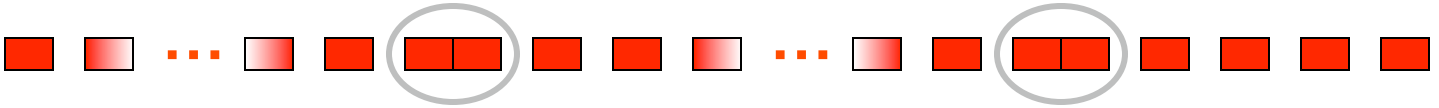
UDT

UDT

- UDT: UDP based Data Transfer
  - Application level transport protocol, over UDP with reliability, congestion, and flow control
  - Implementation: Open source C++ library
- Rate based congestion control (Rate Control)
  - RC tunes the packet sending period.
  - RC is triggered periodically.
- Window based flow control (Flow Control)
  - FC limits the number of unacknowledged packets.
  - FC is triggered on each received ACK.

UDT

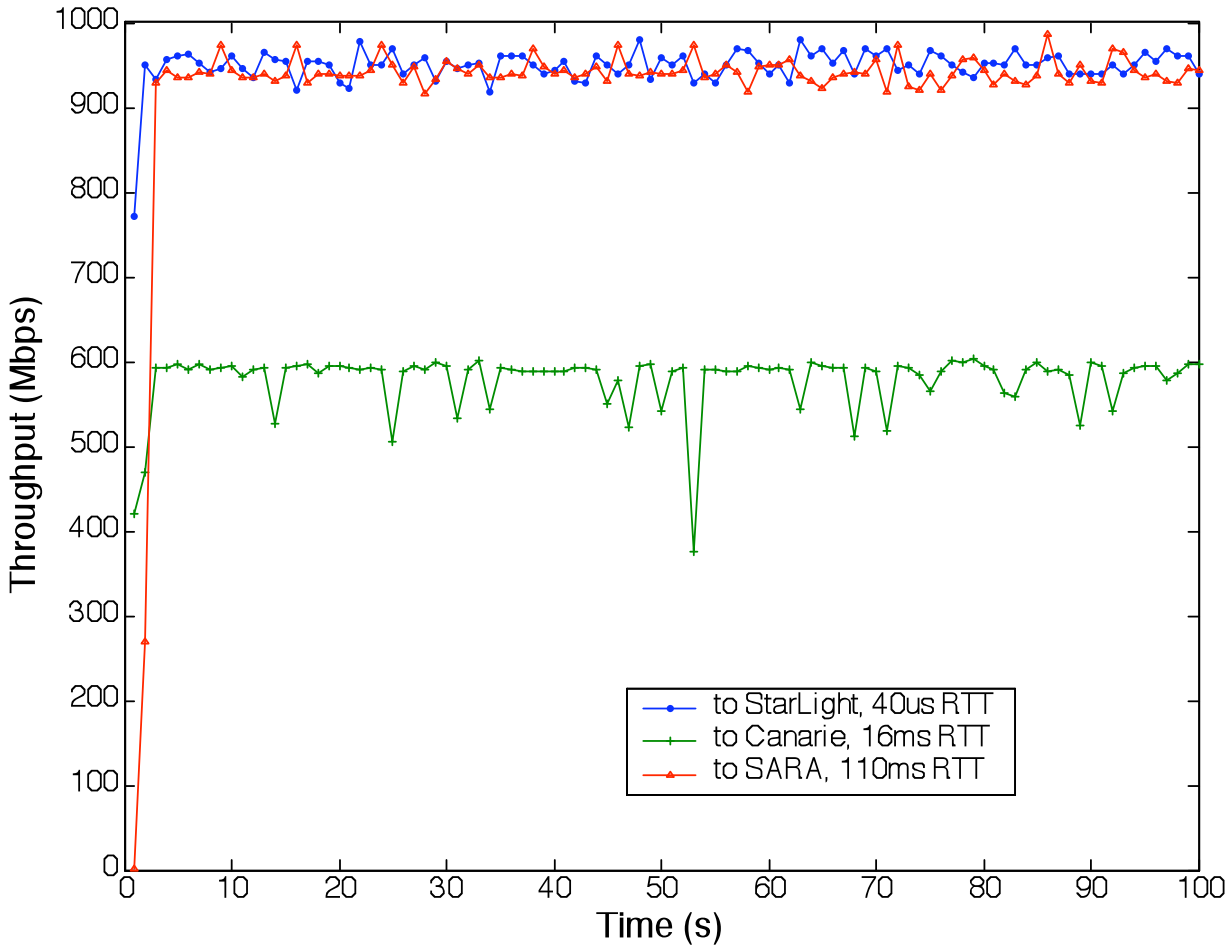
- AIMD: Increase parameter is related to link capacity and current sending rate; Decrease factor is 1/9, but not decrease for all loss events.
- Link capacity is probed by packet pair, which is sampled UDT data packets.
  - Every 16th data packet and it successor packet are sent back to back to form a packet pair.



  - The receiver uses a median filter on the interval between the arrival times of each packet pair to estimate link capacity.

the globus alliance
www.globus.org

UDT



GridFTP/UDT Integration

Wrapblock Driver Development

- Easy way to write XIO Drivers
  - Create from third party libraries.
- Blocking API
  - Thread pooling/event callbacks to morph async to sync
  - Recommend threaded builds
- UDT driver developed using the wrapblock feature

Interface functions

- A set of function signatures
  - open/close/read/write implemented by driver
  - cntl() functions for driver specific hooks
  - Wrapped into a structure and registered with Globus XIO
- Calls to these functions are made expecting specific behaviours
  - Ex: the read() interface function should produce some data, and the write() interface function should consume data, etc

Example Interface Functions

```
static
globus_result_t
globus_l_xio_udt_ref_read(
    void *                          driver_specific_handle,
    const globus_xio_iovec_t *      iovec,
    int                             iovec_count,
    globus_size_t *                 nbytes)
{
    globus_result_t                 result;
    xio_l_udt_ref_handle_t *        handle;

    handle = (xio_l_udt_ref_handle_t *) driver_specific_handle;

    *nbytes = (globus_size_t) UDT::recv(
        handle->sock, (char *)iovec[0].iov_base,
        iovec[0].iov_len, 0);
    /* need to figure out eof */
    if(*nbytes <= 0)
    {
        result = GlobusXIOUdtError("UDT::recv failed");
        goto error;
    }

    return GLOBUS_SUCCESS;
error:
    return result;
}
```

```
static
globus_result_t
globus_l_xio_udt_ref_write(
    void *                          driver_specific_handle,
    const globus_xio_iovec_t *      iovec,
    int                             iovec_count,
    globus_size_t *                 nbytes)
{
    globus_result_t                 result;
    xio_l_udt_ref_handle_t *        handle;

    handle = (xio_l_udt_ref_handle_t *) driver_specific_handle;

    *nbytes = (globus_size_t) UDT::send(
        handle->sock, (char*)iovec[0].iov_base,
        iovec[0].iov_len, 0);
    if(*nbytes < 0)
    {
        result = GlobusXIOUdtError("UDT::send failed");
        goto error;
    }

    return GLOBUS_SUCCESS;
error:
    return result;
}
```

Throughput achieved using various transport mechanisms

Transport Mechanism \ Testbed	ANL - NZ	ANL - ISI	BMI - Japan	Japan – ORNL
Iperf – 1 stream	19.7	74.5	59	110
Iperf – 8 streams	40.3	117.0	115.3	592.4
scp – 1 stream	2.96	9.6	3.13	3.14
bbcp mem – 1 stream	-	35.3	5.84	118.2
bbcp mem – 8 streams	-	59.2	11.7	467.25
bbcp disk – 1 stream	-	35.15	5.85	112.6
bbcp disk – 8 streams	-	54.8	11.7	451.3
GridFTP mem TCP – 1 stream	16.4	63.8	79.6	123.3
GridFTP mem TCP – 8 streams	40.2	112.6	222	586.5
GridFTP disk TCP – 1 stream	16.3	59.6	73.6	113.5
GridFTP disk TCP – 8 streams	37.4	102.4	201.6	574.6
GridFTP mem UDT	189.7	426.6	238	382.5
GridFTP disk UDT	187.9	418.3	220.5	380.6
UDT mem	202.3	432.5	246.3	397.2
UDT disk	174.2	398.0	211.6	374.5

the globus alliance
www.globus.org

Impact of concurrent flows

Japan-ORNL testbed

Nonconcurrent flows	GridFTP-TCP	132.1
	GridFTP-UDT	416.8
2 Concurrent TCP flows	GridFTP-TCP	120.1
	GridFTP-TCP	118.6
2 Concurrent UDT flows	GridFTP-UDT	403.2
	GridFTP-UDT	404.6
2 Concurrent flows (1 TCP and 1 UDT)	GridFTP-TCP	122.6
	GridFTP-UDT	404.3
3 Concurrent flows (2 TCP and 1 UDT)	GridFTP-TCP	122.6
	GridFTP-TCP	123.2
	GridFTP-UDT	405.7

BMI-Japan testbed

Nonconcurrent flows	GridFTP-TCP	80.1
	GridFTP-UDT	240.2
2 Concurrent TCP flows	GridFTP-TCP	78.6
	GridFTP-TCP	79.2
2 Concurrent UDT flows	GridFTP-UDT	120.5
	GridFTP-UDT	126.4
2 Concurrent flows (1 TCP and 1 UDT)	GridFTP-TCP	42.4
	GridFTP-UDT	187.6
3 Concurrent flows (2 TCP and 1 UDT)	GridFTP-TCP	26.5
	GridFTP-TCP	34.6
	GridFTP-UDT	164.4

Resource Utilization of UDT vs TCP

- The performance of TCP and UDT comparable on TeraGrid network between ANL and ORNL
  - Both TCP and UDT achieved a throughput around 700 Mbit/s on this testbed.
- The CPU utilization for TCP transfers was in the range of 30–50%, whereas for UDT transfers it was around 80%
- The memory consumption was around 0.2% for TCP and 1% for UDT

Questions